AF
2131
JFW

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of:

Traversat, et al.

Serial No. 09/653,227

Filed: August 31, 2000

For: Message Authentication using Message Gates in a Distributed Computing Environment

§ Group Art Unit: 2131
§
§ Examiner: Chen, Shin Hon
§
§ Atty. Dkt. No.: 5181-64800
§ P4979

CERTIFICATE OF MAILING
37 C.F.R. § 1.8

I hereby certify that this correspondence is being deposited with the U.S. Postal Service with sufficient postage as First Class Mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on the date indicated below:

Robert C. Kowert
Name of Registered Representative

August 1, 2005
Date

Signature

APPEAL BRIEF

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir/Madam:

Further to the Notice of Appeal mailed June 2, 2005, Appellants present this Appeal Brief. Appellants respectfully request that the Board of Patent Appeals and Interferences consider this appeal.

08/04/2005 EFLORES 00000120 501505 09653227
01 FC:1402 500.00 DA

I. REAL PARTY IN INTEREST

As evidenced by the assignment recorded at Reel/Frame 011070/0082, the subject application is owned by Sun Microsystems, Inc., a corporation organized and existing under and by virtue of the laws of the State of Delaware, and now having its principal place of business at 4150 Network Circle, Santa Clara, CA 95054.

II. RELATED APPEALS AND INTERFERENCES

This appeal is related to the pending appeal in U.S. Application No. 09/653,215 in that both inventions pertain to similar technologies and both applications have been rejected by the same Examiner using the same primary prior art reference.

III. STATUS OF CLAIMS

Claims 1-6, 8-31, 33-47 and 49-72 stand finally rejected. The rejection of claims 1-6, 8-31, 33-47 and 49-72 is being appealed. A copy of claims 1-6, 8-31, 33-47 and 49-72 as currently pending is included in the Claims Appendix herein below.

IV. STATUS OF AMENDMENTS

An amendment, submitted via Facsimile on July 28, 2005, amends claims 62-72 to recite a tangible computer accessible medium rather than a carrier medium. The Examiner has not indicated whether this amendment will be entered. The Claims Appendix included herewith reflects the state of the claims prior to this amendment.

V. SUMMARY OF CLAIMED SUBJECT MATTER

Independent claim 1 is directed to a method for communicating in a distributed computing environment including a client accessing an authentication service to obtain an authentication credential to use a service. A client may access or use an authentication service in various ways. A client process may directly communicate with an authentication service to obtain an authentication credential, in some embodiments. In

other embodiments, a gate factory on a client may use an authentication service to obtain a authentication credential to be embedded in messages. In yet other embodiments, a gate factory may create or include its own message gate used to communicate with an authentication service to receive an authentication credential. A client may discover an authentication service from a service advertisement, such as may be stored in a network addressable space service. The advertisement may include an address, such as a URI, for accessing the authentication service. A client may present a client identification token or other information as proof of the client's identity to an authentication service. The authentication service may issue to the client an authentication credential that only the authentication service can create. While in some embodiments an authentication credential may be unique to the particular client, in other embodiments, the credential may be a prearranged credential that all clients of a particular service are to use. *See, e.g.*, Figs. 26a, 26b, 41, 42b, 42c and 43; p. 13, line 31 – p. 14, line 4; p. 33, lines 5-16; p. 35, line 22 – p. 36, line 4; p. 63, lines 3-12; p. 63, line 24 – p. 64, line 4; p. 66, line 22 – p. 67, line 9; p. 69, lines 8 – 21; p. 84, lines 6-21; p. 86, lines 11-28; p. 91, line 25 – p. 92, line 2; p. 94, lines 1- 21; p. 96, line 20 – p. 97, line 26; p. 98, line 4 – p. 99, line 2; and p. 99, line 6 – p. 100, line 27.

The method also includes determining client capabilities for the client. The client capabilities are capabilities of the service that the client is permitted to use. A distributed computing environment may include a mechanism for client to negotiate access rights to use a services capabilities or a subset of a service's full capabilities. The result of such negotiation may be an authentication credential that conveys to the client the right to use some or all of a service's capabilities. In one embodiment, information received in a request message may be used to determine the capabilities of the client to use a service. In some embodiments an authentication service, such as one used to obtain an authentication credential, may determine the capabilities of the client upon receiving a client's authentication credential from a service desiring to verify the client's authentication. In other embodiments, the service itself may determine the specific capabilities that a client is allowed to use. The method further includes binding the client capabilities to the authentication credential. In one embodiment, the service may bind the

client's capabilities to the authentication credential. *See, e.g.*, Figs. 26a, 26b, 41, 42b, 42c and 43; p. 56, lines 4 – 21; p. 63, lines 14 – 21; p. 64, lines 2-11; p. 67, lines 4 – 9; p. 76, lines 1 – 9; p. 85, lines 2 – 11; p. 92, line 20 – p. 93, line 2; and p. 94, lines 4 – 21.

The client sends a message including the authentication credential to the service and the service uses the authentication service to authenticate the authentication credential received in the message. Credentials may be used to verify the identity and/or rights of a client to use a service. In one embodiment, an authentication credential may be presented each time a client uses a service. In some embodiments, a message gate for a client may present the authentication credential. The service receiving the authentication credential may use the authentication credential to ensure that the authentication credential is valid and belongs to the client. By using the authentication service to authenticate the client, the service may establish a binding of the authentication credential to the identity of the client. The sharing a single authentication services by both a client and service, any variety of authentication protocols may be employed, with the details of the particular authentication protocol being separated from both the client and the service. The service responds to the message if the authentication credential in the message is determined to be authentic as from the client. *See, e.g.*, Figs. 26a, 26b, 41, 42b, 42c and 43; p. 13, line 28 – p. 14, line 11; p. 32, line 28 – p. 33, line 16; p. 67, lines 4 – 14; p. 84, lines 23- 30; p. 85, lines 2 – 16; p. 87, line 1 – p. 88, line 27; p. 91, line 25 – p. 92, line 2; p. 92, line 20 – p. 93, line 2; p. 93, line 28 – p. 94, line 21; p. 96, line 20 – p. 99, line 2; and p. 105, line 19 – p. 107, line 27.

Independent claim 17 is directed to a method for communication is a distributed computing environment in which a client obtains a service advertisement for a service. An advertisement may provide a mechanism of addressing and accessing services and/or content within the distributed computing environment. Services in a distributed computing environment may publish, such as on a space service, an advertisement for the service. An advertisement may be represented in XML and may include a message schema and an address for accessing the service. Clients may search for or browse published advertisements. Advertisements may be complete advertisements that include

a message schema or interface for accessing the service or may be protected advertisements that don't include such a schema or interface. Service advertisement may also include an address for an authentication service that the client may use to obtain an authentication credential and that the service may use to authenticate the client. *See, e.g.*, Figs. 4, 8, 9, 11a, 15, 16, 17, 18, 21, 22, 23, 24, 25, 27, 28, 29, 31, 32A, 32B, 36, 38 and 39a; p. 27, lines 13 – 22; p. 28, lines 5 – 16; p. 28, line 26 – p. 29, line 7; p. 29, lines 13 – 23; p. 44, lines 16 – 25; p. 45, lines 10-20; p. 55, line 25 – p. 56, line 2; p. 56, lines 10 – 30; p. 57, line 19 – p. 58, line 8; p. 59, line 5 – p. 61, line 19; p. 62, line 10 – 63, line 12; p. 68, lines 7-26; p. 74, line 2 – p. 75, line 11; p. 75, line 24 – p. 76, line 9; p. 79, line 17- p. 80, line 9; p. 84, lines 6 – 21; p. 86, line 11 – 28; and p. 98, line 28 – p. 99, line 21.

The client sends a request message to the authentication service to obtain an authentication credential to use the service. Please refer to the discussion of claim 1 above for more information regarding a client obtaining an authentication credential to use a service.

The client also generates a message gate for accessing the service. Message gates may provide secure message endpoints in a distributed computing environment. A pair of message gates may provide a mechanism for communicating requests from client to services and responses from services to clients. Two associated message gates may be used to create a secure atomic bi-directional messaging channel for request-response message passage. Messages gates may allow clients and services to exchange messages in a secure and reliable fashion over any suitable message transport (e.g. HTTP). For a client, a message gate may represent the authority to use some or all of a service's capabilities. In one embodiment, message gates may be created that may only send and/or receive a subset of the total message schema for a service. The message gates may perform verification of the messages against the data representation language message schema to ensure that the message is in the allowed subset of messages. Each message may also include a token or credential that includes information that may allow the receiving gate to verify that the message has not been compromised or altered. A distributed computing environment may include several different types of messages gates

for communicating between clients and services. Some gates may support flow control while other gates may support remote method invocation. Other gates may support publish and subscribe message passing for events. Message gates may be created from information, such as a message schema, in an advertisement for a service. Message gates may also incorporate an authentication credential obtained from an authentication service. *See, e.g.*, Figs. 10 - 15, 20, 22, 25, 34, 35a, 41, 42a and 42b; p. 35, line 22 – p. 36, line 4; p. 29, line 27 – p. 44, line 14; p. 47, line 11 – 50, line 3; p. 50, line 24 – p. 52, line 3; p. 53, lines 16 – 29; p. 61, line 27 – p. 62, line 8; p. 84, lines 6 – 30; p. 92, line 15 – p. 93, line 26; p. 97, lines 17 – 26; p. 102, lines 5 – 23; and p. 103, lines 5 – 28.

The message gate may embed the authentication credential in every message from the client to the service. Please refer to the discussion of claim 1 above for more information regarding a message gate embedding an authentication credential in every message from a client to a service.

Independent claim 27 recites a client device configured to access an authentication service to obtain an authentication credential to use a service. Please refer to the discussions of claims 1 and 17 above for more information regarding a client obtaining an authentication credential.

The client device of claim 27 is also configured to determine client capabilities for the client device and may also bind the client capabilities to the authentication credential. Please refer to discussions of claims 1 and 17 for more information regarding determining client capabilities and binding client capabilities to authentication credentials.

The client device of claim 27 is further configured to send a message including the authentication credential to the service. The service is configured to use the authentication service to authenticate the authentication credential received in the message. The client device also receives a response to the message from the service if the authentication credential in the message is determine to be authentic as from the client

device. Please refer to the discussions of claim 1 and 17 above for more details regarding clients including authentication credentials in messages to services and regarding services using authentication services to authenticate received authentication credentials.

Independent claim 43 is directed to a service device configured to receive from a client a message including an authentication credential which the obtained by accessing an authentication service. The service device uses the authentication service to authenticate the authentication credential received in the message and determines client capabilities for the client. The service device is additionally configured to bind the client capabilities to the authentication credential and respond to the first message if the authentication credential in the message is determined to be authentic as from the client. Please refer to the discussions of claims 1 and 17 above for more details regarding services receiving and authenticating authentication credentials from clients and regarding binding client capabilities to authentication credentials.

Independent claim 51 is directed to a distributed computing system including a client device and a service device. The client device is configured to access an authentication service to obtain an authentication credential to use the service device, determine client capabilities for the client device, bind the client capabilities to the authentication credential, and send a message including the authentication credential to the service device. The service device is configured to use the authentication service to authenticate the authentication credential received in the message and respond to the message if the authentication credential in the message is determined to be authentic as from the client device. Please refer to the above discussions regarding the other independent claims for more information regarding the individual features and functionality recited in claim 51.

Independent claim 58 recites a system including a client device and a service device in which the client device and service device implement the method described above regarding claim 17. Please see the above discussion of claim 17 for more details.

Independent claim 62 recites a medium including program instructions that are computer-executable to implement the method described above regarding claim 1. Please see the above discussion of claim 1 for more details.

Independent claim 69 recites a medium including program instructions that are computer-executable to implement the method described above regarding claim 17. Please see the above discussion of claim 17 for more details.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1. Claims 1-6, 8-15, 17-21, 24-31, 33-37, 41-47, 49-55 and 57-72 stand finally rejected under 35 U.S.C. § 102(a) as being anticipated by Czerwinski, et al., "An Architecture for a Secure Service Discovery Service" (hereinafter "Czerwinski").

2. Claims 16, 22, 23, 38, 39 and 56 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Czerwinski in view of Johnson et al. (U.S. Patent 5,560,008) (hereinafter "Johnson").

3. Claim 40 stands finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Czerwinski in view of Appellant's Applied Prior Art.

VII. ARGUMENT

First Ground of Rejection

Claims 1-6, 8-15, 17-22, 24-31, 33-37, 41-47, 49-55 and 57-72 stand finally rejected under 35 U.S.C. § 102(a) as being anticipated by Czerwinski. Appellants traverse this rejection for at least the following reasons. Different groups of claims are addressed under their respective subheadings. Please note that in the final Office Action, the Examiner lists claim 22 under this ground of rejection, but describes the actual rejection of claim 22 under the second ground of rejection. Appellants assume the Examiner intended to reject claim 22 over Czerwinski in view of Johnson. As such,

Appellants address the rejection of Claim 22 under the second ground of rejection, discussed below.

Claims 1, 2, 8, 11, 15, 62, 63 and 66:

Regarding claim 1, Czerwinski fails to disclose binding the client capabilities to the authentication credential. In contrast, Czerwinski teaches two separate mechanisms for authentication credentials and capabilities, respectively. Specifically, Czerwinski discloses a Certificate Authority responsible for providing authentication certificates and a separate Capability Manager that generates and distributes capabilities. Under Czerwinski's system, a client contacts a Capability Manager to obtain a capability credential that is later used when sending a query to discover services. In Czerwinski, the SDS server is responsible for matching a client's query with service descriptions and uses the client's capabilities to ensure that only those services to which the client is granted access are returned to the client. Additionally, a client in Czerwinski's system separately contacts the Certificate Authority to obtain an authentication credential (Czerwinski, sections 3.1, 3.4, and 3.1 paragraph 5). Nowhere does Czerwinski describe binding the capabilities obtained from the Capability Manager with the authentication credential obtained from the Certificate Authority.

In response Appellants' arguments above, the Examiner states, "Czerwinski discloses the CM generates the capability after authentication so that the SDS can perform access control based on the capabilities sent by client" and further states, "the capabilities received by the client from CM is actually the authentication credential." The Examiner's interpretation of Czerwinski is incorrect. Czerwinski specifically states, "[c]apability distribution itself can be done *without authentication* because capabilities, like certificates, are securely associated with a single principal, and only the clients possessing the appropriate private key can use them" (emphasis added, Czerwinski, section 3.4, paragraph 3). Thus, contrary to the Examiner's assertion, a client in Czerwinski's system is not authenticated before receiving a capability. A capability may be associated with a client, without being bound to any authentication credential.

Furthermore, contrary to the Examiner's assertion, the capabilities received by the clients in Czerwinski are **not** authentication credentials.

Claim 3:

Regarding claim 3, contrary to the Examiner's assertion, Czerwinski does not teach that the advertisement for the first service includes a data representation language schema *defining a message interface* for accessing the first service. In contrast, Czerwinski discloses domain advertisements that contain "the multicast address to use for sending service announcements, the desired service announcement rate, and contact information for the Certificate Authority and the Capability Manager" (Czerwinski, section 3.1, paragraph 1). Additionally, Czerwinski's service descriptions contain service metadata, such as location, required capabilities, time-out period, and JAVA RMI addresses (Czerwinski, section 2.3, paragraph 3). Neither the domain advertisements nor the service descriptions of Czerwinski include a data representation language schema defining a message interface for accessing a service.

The Examiner cites a portion of Czerwinski (section 3.1) that describes how a client submits a query in the form of an XML template. However, a client query using an XML template as the content of a query is very different from a data representation language schema defining a message interface for accessing a service. The XML template in a client query in Czerwinski does not define a message interface for accessing a service. Instead client queries include desired services and are matched against service descriptions to find services providing those desired services (Czerwinski, section 2.3, paragraph 3 and section 3.1, paragraph 5). Further, Czerwinski teaches the use of Authenticated Remote Method Invocation (ARMI) for communication between client applications and SDS servers, *and it is well known that ARMI uses Java interface classes, and not data representation language schemas*, to define the methods that are exposed for remote calling. Thus, Czerwinski clearly fails to teach wherein the advertisement for the first service includes a data representation language schema defining a message interface for accessing the first service.

In response the Appellants arguments above, the Examiner argues, "Czerwinski discloses that a client submits a query in the form of an XML template so that the SDS can use the form to perform service [sic] for the client" and further contends, "the XML template is a data representation language schema to define a interface for the service to perform tasks." Appellants strongly disagree. Firstly, the Examiner appears to be ignoring the specific wording of Appellants' claim. Claim 3 recites, in part, a data representation language schema defining a *message* interface for accessing a service. The Examiner has neglected to cite any passage of Czerwinski that discloses a data representation language schema defining a message interface. The Examiner refers to a query in XML, which Czerwinski describes as including such factors as, "cost, performance, location, and device- or service-specific capabilities" (Czerwinski, Abstract). As is readily apparent from Figure 2 of Czerwinski, a client query is not a data representation schema, and clearly does not define a message interface for accessing a service. Furthermore, there is no way for a client to define such a message interface in a query template when the client has not even located a service (that is purpose of submitting the query template). It would be impossible under Czerwinski's teachings for the client to define a message interface for a service that has not even been located and/or selected.

Additionally, the Examiner's argument fails to take into account the language of claim 3 that the *advertisement for the service* includes a data representation language schema defining a message interface for accessing the service. As noted above, the Examiner contends that Czerwinski's client query is such a data representation language schema. However, a client query in Czerwinski's system is not part of any advertisement for a service. Instead, a client submits a query to find a service. Czerwinski teaches that the SDS server uses a search engine to search for service descriptions that match the client query.

Claim 4:

Regarding claim 4, Czerwinski fails to disclose that the first message, sent from the client to the service and including the authentication credential, corresponds to a message defined in the data representation language schema. The Examiner cites page 27, section 3.1 of Czerwinski and argues, "a client submits a query in the form of an XML template." However, regarding claim 3, discussed above, the Examiner argues that the query submitted by a client in the form of an XML template "is a data representation language schema to define a[n] interface for the service to perform tasks." Thus, the Examiner is arguing that a client query in Czerwinski is both a data representation language schema and a message defined in the same data representation language schema. Such and interpretation is not only incorrect, but clearly inconsistent. The client query in Czerwinski cannot be both a data representation language schema (as the Examiner asserts for claim 3) and a message defined in the same data representation language schema (as the Examiner asserts for claim 4).

Additionally, as described above regarding claim 3, a client query in Czerwinski is not a data representation schema, and does not define a message interface for accessing a service. As noted above, there is no way in Czerwinski for a client to define such a message interface in a query template when the client has not even located a service (that is purpose of submitting the query template) and it would be impossible in Czerwinski for the client to define a message interface for a service that has not even been located and/or selected.

Furthermore, Czerwinski teaches the use of Authenticated Remote Method Invocation (ARMI) for communication between client applications and SDS servers, *and it is well known that ARMI uses Java interface classes, not data representation language schemas*, to define the methods that are exposed for remote calling. Thus, the clients do not use message defined in data representation language schemas in Czerwinski's system and certainly do not use messages defined in data representation language schema for submitting queries to SDS servers.

Claims 5 and 6:

Regarding claim 5, Czerwinski fails to disclose the client sending additional messages to the service to use the service wherein the authentication credential is included with each one of the additional messages, and wherein each one of the additional messages is defined by the data representation language schema. The Examiner cites section 3.1 of Czerwinski and refers to how a client in Czerwinski uses Authenticated RMI (ARMI) to connect to an SDS server. However, as discussed above regarding claims 3 and 4, it is well known that ARMI uses Java interface classes to define methods that are exposed for remote calling. ARMI does not use data representation language schemas to define exposed methods for remote calling.

Additionally, Czerwinski teaches that authentication in ARMI "consists of a short handshake that establishes a symmetric [encryption] key used for the rest of the session" and that "ARMI uses certificates to authenticate each of the endpoints" (Czerwinski, page 28, section 3.5.3). Thus, Czerwinski teaches performing a handshake once at the beginning of a session in which certificates are used to authenticate each endpoint and the symmetric encryption key is used for the remainder of the session. Czerwinski does not mention including an authentication credential with each additional message. Since any additional messages (after the initial handshake) are encrypted and decrypted using the symmetric encryption key, there is not need to include any authentication credential with each message. Hence, **Czerwinski teaches away** from an authentication credential included with each one of the additional messages.

Claim 9:

Regarding claim 9, Czerwinski does not disclose that determining client capabilities comprises the client accessing an access policy service to obtain a capability token indicating which capabilities of the first service the client is permitted to access. The Examiner cites page 28, section 3.4 describing Czerwinski's Capability Manager. However, as described previously, the Capability Manager issues capabilities that indicate whether a client is allowed to access the service's service description (Czerwinski, section 3.4, paragraph 3). Thus, the token issued by Czerwinski's

Capability Manager does not indicate which capabilities of a service a client is permitted to access. Instead, the capability obtained from the Capability Manager only indicates whether or not the client is allowed to access service descriptions which are not capabilities of the service that the client is permitted to access. Czerwinski teaches that a capability "proves that a particular client is on the access control list for a service by embedding the client's principal name and the service name" (page 28, section 3.4, paragraph 2). Czerwinski does not teach that a capability issued by the Capability Manager indicates which capabilities of a service the client is permitted to access.

Claim 10:

Regarding claim 10, Czerwinski does not disclose that the authentication service and the access policy service are combined as a single service. The Examiner cites page 28, section 3.4 describing Czerwinski's Capability Manager. The Capability Manager relied upon by the Examiner cannot be considered an authentication service. Czerwinski's Capability Manager does not perform any sort of authentication service. Czerwinski clearly states, "[c]apability distribution can be done without authentication because capabilities, like certificates, are securely associated with a single principal, and only clients possessing the appropriate private key can use them" (emphasis added, page 28, section 3.4, paragraph 3). Additionally, Czerwinski teaches a Certificate Authority (page 27, section 3.3) that signs authentication certificates for principals in Czerwinski's system. A client in Czerwinski contact the Certificate Authority to obtain authentication and encryption certificates. The Capability Manager cited by the Examiner and the Certification Authority are not combined as a single service in Czerwinski's system.

Furthermore, Czerwinski does not disclose that the capability token is included within the authentication credential. Instead, Czerwinski teaches authentication and encryption certificates obtained from the Certification Authority (see, section 3.3). Neither an authentication certificate nor an encryption certificate includes any sort of capability token in Czerwinski's system. Nowhere does Czerwinski mention a capability token included within an authentication credential.

Claims 12 and 13:

Regarding claim 12, Czerwinski fails to disclose that the message gate includes the authentication credential in each message to the first service. The Examiner cites sections 3.1, 3.3 and 3.4 of Czerwinski. However, none of these passages mentions anything regarding a message gate including an authentication credential in each message to a service. Czerwinski's system does not include authentication credentials in each message to a service. Instead, as described above, Czerwinski teaches the use of Authentication RMI (ARMI) in which a handshake is performed at the start of a client session. During the handshaking, a symmetric encryption key is established and used to encrypt messages for the rest of the session. (See, page 28, section 3.5.3). When using ARMI there is not need to include an authentication credential in each message to a service. Czerwinski does not teach or describe anything about including an authentication credential in each message to a service.

Claim 14:

Regarding claim 14, contrary to the Examiner's assertion, Czerwinski does not disclose that the advertisement for the first service includes a data representation language schema *defining a message interface* for accessing the first service. In contrast, Czerwinski discloses domain advertisements that contain "the multicast address to use for sending service announcements, the desired service announcement rate, and contact information for the Certificate Authority and the Capability Manager" (Czerwinski, section 3.1, paragraph 1). Additionally, Czerwinski's service descriptions contain service metadata, such as location, required capabilities, time-out period, and JAVA RMI addresses (Czerwinski, section 2.3, paragraph 3). Neither the domain advertisements nor the service descriptions of Czerwinski include a data representation language schema defining a message interface for accessing a service. For a more detailed discussion regarding Czerwinski's failure to teach including, in a service advertisement, a data representation language schema defining a message interface for accessing a service, please see the discussion of claim 3 above.

Additionally, Czerwinski does not disclose wherein the message gate verifies that each message sent from the client to the first service complies with the data representation language schema. The Examiner again cites sections 3.1, 3.3 and 3.4 of Czerwinski. However, none of these sections makes any reference whatsoever to any sort of message gate verifying that messages sent from a client to a service comply with a data representation language schema. The Examiner has not provided any explanation regarding what portion or entity of Czerwinski is relied upon to be a message gate that verifies that message comply with a data representation language schema. Furthermore, as described previously, Czerwinski teaches the use of Authenticated RMI that, as noted above, does not use data representation language schemas and thus does not include any message gate verifying that messages comply with a data representation language schema (See, page 28, section 3.5.3).

Claims 17, 20, 21 and 24:

Regarding claim 17, contrary to the Examiner's assertion, Czerwinski fails to teach the client generating a message gate for accessing the first service, wherein the message gate embeds the authentication credential in *every* message from the client to the first service. The Examiner has cited sections 3.1, 3.3 and 3.4 of Czerwinski that describe the working of the SDS server, Certificate Authority, and Capability Manager of Czerwinski's system. However, neither the Examiner's cited passage, nor any other portion of Czerwinski, discloses generating a message gate for access the first service, wherein the message gate embeds the authentication credential in every message from the client to the first service. In contrast, Czerwinski teaches the use of Authenticated Remote Method Invocation (ARMI) for communication between client applications and SDS servers. Czerwinski also teaches that ARMI uses certificates to authenticate each of the endpoints and states that such authentication consists of a short handshake to establish a symmetric key used for the rest of the session. (Czerwinski, section 3.5.3, paragraph 2). After the certificate-based authentication at the start of an ARMI session, only the session encryption key is used to validate the remaining messages of the ARMI session.

No certificate or credential is embedded with every message of an ARMI session. Thus, Czerwinski fails to teach embedding the authentication credential in every message from the client to the service.

In response the above arguments, the Examiner states, "Czerwinski discloses that when the client sends a query to the SDS server, the client include[s] the client's capabilities." However, as noted above, Czerwinski teaches that capabilities are distributed without authentication (Czerwinski, section 3.4, paragraph 3). Also, Czerwinski's system specifically includes a Certificate Authority responsible for generating authentication certificates, which are different than capabilities. Thus, Czerwinski's capabilities are clearly not authentication credentials. A client submitting a query to an SDS service and including the client's capabilities does not involve or imply embedding an authentication credential in every message.

Furthermore, the Examiner has failed to rebut Appellants' argument regarding Czerwinski's use of ARMI and that ARMI messaging does not include embedding an authentication credential in every message.

Claims 18 and 19:

Regarding claim 18, contrary to the Examiner's assertion, Czerwinski does not teach that the advertisement for the first service includes a data representation language schema *defining a message interface* for accessing the first service. In contrast, Czerwinski discloses domain advertisements that contain "the multicast address to use for sending service announcements, the desired service announcement rate, and contact information for the Certificate Authority and the Capability Manager" (Czerwinski, section 3.1, paragraph 1). Additionally, Czerwinski's service descriptions contain service metadata, such as location, required capabilities, time-out period, and JAVA RMI addresses (Czerwinski, section 2.3, paragraph 3). Neither the domain advertisements nor the service descriptions of Czerwinski include a data representation language schema defining a message interface for accessing a service. For a more detailed discussion

regarding Czerwinski's failure to teach including, in a service advertisement, a data representation language schema defining a message interface for accessing a service, please see the discussion of claim 3 above.

Additionally, Czerwinski does not disclose the message gate verifies that each message sent from the client to the first service complies with the data representation language schema. The Examiner again cites sections 3.1 of Czerwinski and refers to a client (in Czerwinski) using Authenticated RMI. However, the cited section does not mention any sort of message gate verifying that messages sent from a client to a service comply with a data representation language schema. Please refer to the discussion of claim 14 for a more detailed discussion of Czerwinski's failure to teach a message gate that verifies that messages comply with a data representation language schema.

Claim 25:

Regarding claim 25, Czerwinski fails to disclose that the service advertisement further includes a service identifier token for the first service, wherein the client sending a request message to the authentication service to obtain an authentication credential comprises sending the service identifier token and a client identifier token to the authentication service. The Examiner cites page 28, section 3.4 of Czerwinski and refers to how the Capability Manager issues a capability that "proves that a particular client is on the access control list for a service by embedding the client's principal name and the service name, signed by some well-known authority."

Firstly, the Capability Manager is not an authentication service and thus a client does not send a request message to obtain an authentication credential from the Capability Manager. A client requesting a capability from Czerwinski's Capability Manager cannot be said to disclose the client sending a request message *to the authentication service* to obtain an authentication credential. Secondly, it is Czerwinski's Capability Manager that embeds the client and service names in a capability and issues that capability to the client, which is very different from a sending a service identifier

token and a client identifier token to an authentication service (see section 3.4 of Czerwinski).

Czerwinski's teaching of a Capability Manager issuing to a client a capability with embedded client and service names clearly cannot be considered to anticipate sending the service identifier token and a client identifier token to an authentication service as part of the client sending a request message to the authentication service to obtain an authentication credential.

Claim 26:

Regarding claim 26, Czerwinski does not anticipate an authentication service that generates an authentication credential from a client identifier token and a service identifier token. The Examiner cites page 28, section 3.4 of Czerwinski and refers to "binding the principal name and the service name and signed by some well known authority". However, as described above regarding claim 25, the Capability Manager is not an authentication service and does not generate authentication credentials. Instead, Czerwinski teaches a Certificate Authority responsible for issuing authentication and encryption key certificates. Moreover, Czerwinski's Certificate Authority does not generate authentication credentials from a client identifier token and a service identifier token. Instead, Czerwinski states that a client contacts the Certificate Authority and specifies the principal's (e.g. a service's) certificate that the client is interested in, and the Certificate Authority returns the matching certificate. Czerwinski makes no reference whatsoever to generating authentication credentials from client identifier tokens and a service identifier tokens.

Claims 27, 28, 33 and 41:

Regarding claim 27, Czerwinski fails to teach a *client device* configured to determine client capabilities for the client, wherein the client capabilities are capabilities of the service device that the client is permitted to use, and further fails to teach a *client device* configured to bind the client capabilities to the authentication credential. Instead,

as discussed above, Czerwinski teaches two separate mechanisms for authentication credentials and capabilities. Specifically, Czerwinski discloses a Certificate Authority responsible for providing authentication certificates and a separate Capability Manager that generates and distributes capabilities. None of the clients described by Czerwinski are configured to determine capabilities for a client and bind those capabilities to an authentication credential. In Czerwinski, the Capability Manager then generates capability credentials that are supplied by clients when querying the SDS server to find services. The SDS server ensures that only those services that the client is allowed to discover, based on the client's capability credential, are returned to the client. (Czerwinski, sections 3.1, 3.4, and section 3.1, paragraph 5). Czerwinski does not mention anything about a client device configured to determine client capabilities for the client, wherein the client capabilities are capabilities of the service device that the client is permitted to use. Czerwinski further fails to disclose a client device configured to bind the client capabilities to the authentication credential.

Claims 29 and 31:

Regarding claim 29, contrary to the Examiner's assertion, Czerwinski does not teach that the advertisement for the first service includes a data representation language schema *defining a message interface* for accessing the first service. In contrast, Czerwinski discloses domain advertisements that contain "the multicast address to use for sending service announcements, the desired service announcement rate, and contact information for the Certificate Authority and the Capability Manager" (Czerwinski, section 3.1, paragraph 1). Additionally, Czerwinski's service descriptions contain service metadata, such as location, required capabilities, time-out period, and JAVA RMI addresses (Czerwinski, section 2.3, paragraph 3). Neither the domain advertisements nor the service descriptions of Czerwinski include a data representation language schema defining a message interface for accessing a service. For a more detailed discussion regarding Czerwinski's failure to teach including, in a service advertisement, a data representation language schema defining a message interface for accessing a service, please see the discussion of claim 3 above.

Claim 30:

Regarding claim 30, Czerwinski fails to disclose the client sending additional messages to the service to use the service wherein the authentication credential is included with each one of the additional messages, and wherein each one of the additional messages is defined by the data representation language schema. The Examiner cites page 27, section 3.1 of Czerwinski and refers to how a client in Czerwinski uses Authenticated RMI (ARMI) to connect to an SDS server. However, as discussed above regarding claims 3 and 4, it is well known that ARMI uses Java interface classes to define methods that are exposed for remote calling. ARMI does not use data representation language schemas to define exposed methods for remote calling.

Additionally, Czerwinski teaches that authentication in ARMI "consists of a short handshake that establishes a symmetric [encryption] key used for the rest of the session" and that "ARMI uses certificates to authenticate each of the endpoints" (Czerwinski, page 28, section 3.5.3). Thus, Czerwinski teaches performing a handshake once at the beginning of a session in which certificates are used to authenticate each endpoint and the symmetric encryption key is used for the remainder of the session. Czerwinski does not mention including an authentication credential with each additional message. Since any additional messages (after the initial handshake) are encrypted and decrypted using the symmetric encryption key, there is not need to include any authentication credential with each message. As described above regarding claim 8, **Czerwinski teaches away** from an authentication credential included with each one of the additional messages.

Claim 34:

Regarding claim 34, Czerwinski does not disclose wherein determining client capabilities comprises the client accessing an access policy service to obtain a capability token indicating which capabilities of the first service the client is permitted to access. The Examiner cites page 28, section 3.4 describing Czerwinski's Capability Manager.

However, as discussed above, the Capability Manager issues capabilities that indicate whether a client is allowed to access the service's service description (Czerwinski, section 3.4, paragraph 3). Thus, the token issued by Czerwinski's Capability Manager does not indicate which capabilities of a service a client is permitted to access. Instead, the capability obtains from the Capability Manager only indicates whether or not the client is allowed to access service descriptions which are not capabilities of the service that the client is permitted to access. Czerwinski teaches that a capability "proves that a particular client is on the access control list for a service by embedding the client's principal name and the service name" (page 28, section 3.4, paragraph 2). Czerwinski does not teach that a capability issued by the Capability Manager indicates which capabilities of a service the client is permitted to access.

Claim 35:

Regarding claim 35, Czerwinski does not disclose that the authentication service and the access policy service are combined as a single service. The Examiner cites page 28, section 3.4 describing Czerwinski's Capability Manager. The Capability Manager relied upon by the Examiner cannot be considered an authentication service. Czerwinski's Capability Manager does not perform any sort of authentication service. Czerwinski clearly states, "[c]apability distribution can be done without authentication because capabilities, like certificates, are securely associated with a single principal, and only clients possessing the appropriate private key can use them" (emphasis added, page 28, section 3.4, paragraph 3). Additionally, Czerwinski teaches a Certificate Authority (page 27, section 3.3) that signs authentication certificates for principals in Czerwinski's system. A client in Czerwinski contact the Certificate Authority to obtain authentication and encryption certificates. The Capability Manager cited by the Examiner and the Certification Authority are not combined as a single service in Czerwinski's system.

Furthermore, Czerwinski does not disclose that the capability token is included within the authentication credential. Instead, Czerwinski teaches authentication and encryption certificates obtained from the Certification Authority. Neither an

authentication certificate nor an encryption certificate includes any sort of capability token in Czerwinski's system. Nowhere does Czerwinski mention a capability token included within an authentication credential.

Claim 36:

Regarding claim 36, Czerwinski fails to disclose wherein the message gate includes the authentication credential in each message to the first service. The Examiner cites sections 3.1, 3.3 and 3.4 of Czerwinski. However, none of these passages mentions anything regarding a message gate including an authentication credential in each message to a service. Czerwinski's system does not include authentication credentials in each message to a service. Instead, Czerwinski teaches the use of Authentication RMI (ARMI) in which a handshake is performed at the start of a client session. During the handshaking, a symmetric encryption key is established and used to encrypt messages for the rest of the session. (See, page 28, section 3.5.3). When using ARMI there is not need to include an authentication credential in each message to a service. Czerwinski does not teach or describe anything about including an authentication credential in each message to a service.

Claim 37:

Regarding claim 37, contrary to the Examiner's assertion, Czerwinski does not teach that the advertisement for the first service includes a data representation language schema *defining a message interface* for accessing the first service. In contrast, Czerwinski discloses domain advertisements that contain "the multicast address to use for sending service announcements, the desired service announcement rate, and contact information for the Certificate Authority and the Capability Manager" (Czerwinski, section 3.1, paragraph 1). Additionally, Czerwinski's service descriptions contain service metadata, such as location, required capabilities, time-out period, and JAVA RMI addresses (Czerwinski, section 2.3, paragraph 3). Neither the domain advertisements nor the service descriptions of Czerwinski include a data representation language schema defining a message interface for accessing a service.

The Examiner cites a portion of Czerwinski (section 3.1) that describes how a client submits a query in the form of an XML template. However, a client query using an XML template as the content of a query is very different from a data representation language schema defining a message interface for accessing a service. The XML template in a client query in Czerwinski does not define a message interface for accessing a service. Instead client queries include desired services and are matched against service descriptions to find services providing those desired services (Czerwinski, section 2.3, paragraph 3 and section 3.1, paragraph 5). Further, Czerwinski teaches the use of Authenticated Remote Method Invocation (ARMI) for communication between client applications and SDS servers, *and it is well known that ARMI uses Java interface classes, and not data representation language schemas*, to define the methods that are exposed for remote calling. Thus, Czerwinski clearly fails to teach wherein the advertisement for the first service includes a data representation language schema defining a message interface for accessing the first service.

Additionally, Czerwinski does not teach the message gate verifies that each message sent from the client to the first service complies with the data representation language schema. The Examiner again cites sections 3.1 of Czerwinski and refers to a client (in Czerwinski) using Authenticated RMI. However, the cited section does not mention any sort of message gate verifying that messages sent from a client to a service comply with a data representation language schema. Furthermore, as discussed above, Authenticated RMI does not use data representation language schemas and thus does not include any message gate verifying that messages comply with a data representation language schema (See, page 28, section 3.5.3).

Claim 42:

Regarding claim 42, Czerwinski fails to disclose wherein the first service is configured to execute within the client device. The Examiner cites page 27, section 3.1 and refers to remote method invocation. However, remote method invocation does not

disclose, teach or even imply that one of Czerwinski's services is configured to execute within the client device. The passage cited by the Examiner does not mention anything about a service executing within a client device. Moreover, Czerwinski's system is specifically designed to allow clients to discover and utilize services that are executing on various machines connected via a network. Czerwinski does not describe any services configured to execute within a client device.

Claims 43, 49 and 50:

Regarding claim 43, Czerwinski fails to teach a *service device* configured to determine client capabilities for the client, wherein the client capabilities are capabilities of the service device that the client is permitted to use, and also fails to teach a *service device* configured to bind the client capabilities to the authentication credential. Instead, as described above, Czerwinski teaches two separate mechanisms for authentication credentials and capabilities. Specifically, Czerwinski discloses a Certificate Authority responsible for providing authentication certificates and a separate Capability Manager that generates and distributes capabilities. None of the services described by Czerwinski are configured to determine capabilities for a client and bind those capabilities to an authentication credential. Instead services in Czerwinski's system contact the Capability Manager and specify those principals, such as clients, that are allowed to discover and access that service. Thus, each of Czerwinski's services supplies an access control list (ACL) to the Capability manager. The Capability Manager then generates and distributes capability credentials that are supplied by clients when querying an SDS server to find services. The SDS server ensures that only those services that the client is allowed to discover, based on the client's capabilities, are returned to the client. (Czerwinski, sections 3.1, 3.4, and section 3.1, paragraph 5). Czerwinski does not mention anything about a *service device* configured to determine client capabilities for the client, wherein the client capabilities are capabilities of the service device that the client is permitted to use, and further fails to mention a *service device* configured to bind the client capabilities to the authentication credential.

Claims 44 and 47:

Regarding claim 44, contrary to the Examiner's assertion, Czerwinski does not teach that the advertisement for the first service includes a data representation language schema *defining a message interface* for accessing the first service. In contrast, Czerwinski discloses domain advertisements that contain "the multicast address to use for sending service announcements, the desired service announcement rate, and contact information for the Certificate Authority and the Capability Manager" (Czerwinski, section 3.1, paragraph 1). Additionally, Czerwinski's service descriptions contain service metadata, such as location, required capabilities, time-out period, and JAVA RMI addresses (Czerwinski, section 2.3, paragraph 3). Neither the domain advertisements nor the service descriptions of Czerwinski include a data representation language schema defining a message interface for accessing a service. For a more detailed discussion regarding Czerwinski's failure to teach including, in a service advertisement, a data representation language schema defining a message interface for accessing a service, please see the discussion of claim 3 above.

Claim 45:

Regarding claim 45, Czerwinski fails to disclose that the first message, sent from the client to the service and including the authentication credential, corresponds to a message defined in the data representation language schema. The Examiner cites page 27, section 3.1 of Czerwinski and argues, "a client submits a query in the form of an XML template." However, regarding claim 3, discussed above, the Examiner argues that the query submitted by a client in the form of an XML template "is a data representation language schema to define a[n] interface for the service to perform tasks." Thus, the Examiner is arguing that a client query in Czerwinski is both a data representation language schema and a message defined in the same data representation language schema. Such and interpretation is not only incorrect, but clearly inconsistent. See the discussion of claim 4 above.

Moreover, as described above regarding claim 3, a client query in Czerwinski is not a data representation schema, and does not define a message interface for accessing a service. As noted above, there is no way for a client to define such a message interface in a query template when the client has not even located a service (that is purpose of submitting the query template) and it would be impossible for the client to define a message interface for a service that has not even been located and/or selected.

Furthermore, Czerwinski teaches the use of Authenticated Remote Method Invocation (ARMI) for communication between client applications and SDS servers, *and it is well known that ARMI uses Java interface classes, and not data representation language schemas*, to define the methods that are exposed for remote calling. Thus, the clients do not use message defined in data representation language schemas in Czerwinski's system and certainly do not use messages defined in data representation language schema for submitting queries to SDS servers.

Claim 46:

Regarding claim 46, Czerwinski fails to disclose the client sending additional messages to the service to use the service wherein the authentication credential is included with each one of the additional messages, and wherein each one of the additional messages is defined by the data representation language schema. The Examiner cites page 27, section 3.1 of Czerwinski and refers to how a client in Czerwinski uses Authenticated RMI (ARMI) to connect to an SDS server. However, as discussed above regarding claims 3 and 4, it is well known that ARMI uses Java interface classes to define methods that are exposed for remote calling. ARMI does not use data representation language schemas to define exposed methods for remote calling.

Additionally, Czerwinski teaches that authentication in ARMI "consists of a short handshake that establishes a symmetric [encryption] key used for the rest of the session" and that "ARMI uses certificates to authenticate each of the endpoints" (Czerwinski, page 28, section 3.5.3). Thus, Czerwinski teaches performing a handshake once at the

beginning of a session in which certificates are used to authenticate each endpoint and the symmetric encryption key is used for the remainder of the session. Czerwinski does not mention including an authentication credential with each additional message. Since any additional messages (after the initial handshake) are encrypted and decrypted using the symmetric encryption key, there is not need to include any authentication credential with each message. As described above regarding claim 8, **Czerwinski teaches away** from an authentication credential included with each one of the additional messages.

Claims 51 and 57:

Regarding claim 51, Czerwinski fails to disclose a *client device* configured to determine client capabilities for the client, wherein the client capabilities are capabilities of the service device that the client is permitted to use, and further fails to teach a *client device* configured to bind the client capabilities to the authentication credential. Instead, as discussed above, Czerwinski teaches two separate mechanisms for authentication credentials and capabilities. Specifically, Czerwinski discloses a Certificate Authority responsible for providing authentication certificates and a separate Capability Manager that generates and distributes capabilities. For a more detailed discussion regarding Czerwinski's failure to teach a client device determining client capabilities and to bind the client capabilities to an authentication credential, please see the discussion of claim 27, above.

Claim 52:

Regarding claim 52, Czerwinski does not disclose a service device configured to provide to the client device an advertisement for the service device, wherein the advertisement includes a data representation language schema defining a message interface for accessing the service device. The Examiner cites sections 2.3 and 3.1 of Czerwinski. However, neither of these cited passages describes an advertisement including a data representation language schema defining a message interface for accessing a service device. Instead, Czerwinski discloses domain advertisements that contain "the multicast address to use for sending service announcements, the desired

service announcement rate, and contact information for the Certificate Authority and the Capability Manager" (Czerwinski, section 3.1, paragraph 1). Additionally, Czerwinski's service descriptions contain service metadata, such as location, required capabilities, time-out period, and JAVA RMI addresses (Czerwinski, section 2.3, paragraph 3). For a more detailed discussion regarding Czerwinski's failure to anticipate an advertisement that includes a data representation language schema defining a message interface for accessing a service device, please refer to the discussions above regarding claims 3, 14, 18, 29, 37 and 44.

Claims 53 and 55:

Regarding claim 53, contrary to the Examiner's assertion, Czerwinski does not disclose that the advertisement for the first service includes a data representation language schema *defining a message interface* for accessing the first service. In contrast, Czerwinski discloses domain advertisements that contain "the multicast address to use for sending service announcements, the desired service announcement rate, and contact information for the Certificate Authority and the Capability Manager" (Czerwinski, section 3.1, paragraph 1). Additionally, Czerwinski's service descriptions contain service metadata, such as location, required capabilities, time-out period, and JAVA RMI addresses (Czerwinski, section 2.3, paragraph 3). Neither the domain advertisements nor the service descriptions of Czerwinski include a data representation language schema defining a message interface for accessing a service. For a more detailed discussion regarding Czerwinski's failure to teach including, in a service advertisement, a data representation language schema defining a message interface for accessing a service, please see the discussion of claim 3 above.

Claim 54:

Regarding claim 54, Czerwinski fails to disclose the client sending additional messages to the service to use the service wherein the authentication credential is included with each one of the additional messages, and wherein each one of the additional messages is defined by the data representation language schema. The

Examiner cites page 27, section 3.1 of Czerwinski and refers to how a client in Czerwinski uses Authenticated RMI (ARMI) to connect to an SDS server. However, as discussed above regarding claims 3 and 4, it is well known that ARMI uses Java interface classes to define methods that are exposed for remote calling. ARMI does not use data representation language schemas to define exposed methods for remote calling.

Additionally, Czerwinski teaches that authentication in ARMI "consists of a short handshake that establishes a symmetric [encryption] key used for the rest of the session" and that "ARMI uses certificates to authenticate each of the endpoints" (Czerwinski, page 28, section 3.5.3). Thus, Czerwinski teaches performing a handshake once at the beginning of a session in which certificates are used to authenticate each endpoint and the symmetric encryption key is used for the remainder of the session. Czerwinski does not mention including an authentication credential with each additional message. Since any additional messages (after the initial handshake) are encrypted and decrypted using the symmetric encryption key, there is not need to include any authentication credential with each message. As discussed above regarding claim 8, **Czerwinski clearly teaches away** from an authentication credential included with each one of the additional messages.

Claims 58 and 61:

Regarding claim 58, Czerwinski fails to disclose a client device configured to generate a message gate for accessing the service device, wherein the message gate is configured to embed the authentication credential in *every* message from the client device to the service device. The Examiner has cited sections 3.1, 3.3 and 3.4 of Czerwinski that describe the working of the SDS server, Certificate Authority, and Capability Manager of Czerwinski's system. However, neither the Examiner's cited passage, nor any other portion of Czerwinski, discloses generating a message gate for access the first service, wherein the message gate embeds the authentication credential in every message from the client to the first service. In contrast, Czerwinski teaches the use of Authenticated Remote Method Invocation (ARMI) for communication between client applications and SDS servers. Czerwinski also teaches that ARMI uses certificates to authenticate each of

the endpoints and states that such authentication consists of a short handshake to establish a symmetric key used for the rest of the session. (Czerwinski, section 3.5.3, paragraph 2). After the certificate-based authentication at the start of an ARMI session, only the session encryption key is used to validate the remaining messages of the ARMI session. No certificate or credential is embedded with every message of an ARMI session. Thus, Czerwinski fails to teach embedding the authentication credential in every message from the client to the service.

For more details regarding Czerwinski's failure to teach a message gate that embeds an authentication credential in every message from a client to a service, please see the above discussion regarding claim 17.

Claims 59 and 60:

Regarding claim 59, contrary to the Examiner's assertion, Czerwinski does not teach that the advertisement for the first service includes a data representation language schema *defining a message interface* for accessing the first service. In contrast, Czerwinski discloses domain advertisements that contain "the multicast address to use for sending service announcements, the desired service announcement rate, and contact information for the Certificate Authority and the Capability Manager" (Czerwinski, section 3.1, paragraph 1). Additionally, Czerwinski's service descriptions contain service metadata, such as location, required capabilities, time-out period, and JAVA RMI addresses (Czerwinski, section 2.3, paragraph 3). Neither the domain advertisements nor the service descriptions of Czerwinski include a data representation language schema defining a message interface for accessing a service. For a more detailed discussion regarding Czerwinski's failure to teach including, in a service advertisement, a data representation language schema defining a message interface for accessing a service, please see the discussion of claim 3 above.

Additionally, Czerwinski does not teach the message gate verifies that each message sent from the client to the first service complies with the data representation

language schema. The Examiner again cites sections 3.1 of Czerwinski and refers to a client (in Czerwinski) using Authenticated RMI. However, the cited section does not mention any sort of message gate verifying that messages sent from a client to a service comply with a data representation language schema. For a more detailed discussion regarding Czerwinski's failure to teach a message that verifies that each message complies with a data representation language schema, please see the discussion of claim 14, discussed above.

Claims 64 and 65:

Regarding claim 64, contrary to the Examiner's assertion, Czerwinski does not teach that the advertisement for the first service includes a data representation language schema *defining a message interface* for accessing the first service. In contrast, Czerwinski discloses domain advertisements that contain "the multicast address to use for sending service announcements, the desired service announcement rate, and contact information for the Certificate Authority and the Capability Manager" (Czerwinski, section 3.1, paragraph 1). Additionally, Czerwinski's service descriptions contain service metadata, such as location, required capabilities, time-out period, and JAVA RMI addresses (Czerwinski, section 2.3, paragraph 3). Neither the domain advertisements nor the service descriptions of Czerwinski include a data representation language schema defining a message interface for accessing a service. For a more detailed discussion regarding Czerwinski's failure to teach including, in a service advertisement, a data representation language schema defining a message interface for accessing a service, please see the discussion of claim 3 above.

Claim 67:

Regarding claim 67, Czerwinski fails to disclose the client generating a message gate for accessing the first service and the message sending request messages from the client to the first service to access the first service, wherein the message gate includes the authentication credential in each message to the first service. The Examiner cites sections 3.1, 3.3 and 3.4 of Czerwinski. However, as discussed above regarding claims

17, 30, 36, 46, and 54, none of the passages cited by the Examiner describes a message that includes an authentication credential in each message to the first service. Instead, Czerwinski utilized the ARMI protocol, which uses authentication credentials only during a handshake at the start of a client session. During the handshake, a symmetric encryption key is established that is used for encrypting all other messages during the client session. Thus, instead of teaching a message gate that includes the authentication credential in each message to a service, Czerwinski teaches including the authentication credential only during an initial handshake between a client and a server. For more a more detailed discussion regarding Czerwinski's failure to disclose a message including an authentication credential in each message to a service, please refer to the discussions of claims 17, 30, 36, 46 and 54 above.

Claim 68:

Regarding claim 68, Czerwinski does not disclose a message gate verifying that each message send from the client to the first service complies with a data representation language schema, wherein the data representation language schema defines a message interface for accessing the first service. The Examiner cites sections 3.1, 3.3 and 3.4 of Czerwinski. However, as discussed previously, none of the cited passage, nor any other portion of Czerwinski, mentions anything about a message gate verifying that messages sent from a client to a service comply with a data representation language schema that defines a message interface for accessing the service. For more a more detailed discussion regarding Czerwinski's failure to disclose a message gates verifying that each message sent from a client to a service complies with a data representation language schema that defines a message interface for accessing the service, please refer to the discussions of claims 14, 18, 37 and 59, above.

Claims 69 and 72:

Regarding claim 69, Czerwinski fails to disclose a client generating a message gate for accessing the first service, wherein the message gate embeds the authentication credential in *every* message from the client to the first service. The Examiner has cited

sections 3.1, 3.3 and 3.4 of Czerwinski that describe the working of the SDS server, Certificate Authority, and Capability Manager of Czerwinski's system. However, neither the Examiner's cited passage, nor any other portion of Czerwinski, discloses generating a message gate for access the first service, wherein the message gate embeds the authentication credential in every message from the client to the first service. In contrast, Czerwinski teaches the use of Authenticated Remote Method Invocation (ARMI) for communication between client applications and SDS servers. Czerwinski also teaches that ARMI uses certificates to authenticate each of the endpoints and states that such authentication consists of a short handshake to establish a symmetric key used for the rest of the session. (Czerwinski, section 3.5.3, paragraph 2). After the certificate-based authentication at the start of an ARMI session, only the session encryption key is used to validate the remaining messages of the ARMI session. No certificate or credential is embedded with every message of an ARMI session. Thus, Czerwinski fails to teach embedding the authentication credential in every message from the client to the service.

For more details regarding Czerwinski's failure to teach a message gate that embeds an authentication credential in every message from a client to a service, please see the above discussion regarding claim 17.

Claims 70 and 71:

Regarding claim 70, contrary to the Examiner's assertion, Czerwinski does not teach that the advertisement for the first service includes a data representation language schema *defining a message interface* for accessing the first service. In contrast, Czerwinski discloses domain advertisements that contain "the multicast address to use for sending service announcements, the desired service announcement rate, and contact information for the Certificate Authority and the Capability Manager" (Czerwinski, section 3.1, paragraph 1). Additionally, Czerwinski's service descriptions contain service metadata, such as location, required capabilities, time-out period, and JAVA RMI addresses (Czerwinski, section 2.3, paragraph 3). Neither the domain advertisements nor the service descriptions of Czerwinski include a data representation language schema

defining a message interface for accessing a service. For a more detailed discussion regarding Czerwinski's failure to teach including, in a service advertisement, a data representation language schema defining a message interface for accessing a service, please see the discussion of claim 3 above.

Second Ground of Rejection

Claims 16, 22, 23, 38, 39 and 56 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Czerwinski in view of Johnson. Appellants traverse this rejection for at least the reasons given above regarding their respective independent claims. Please note that in the final Office Action, the Examiner lists claim 22 under the first ground of rejection, but includes the actual discussion of the rejection of claim 22 under the second ground of rejection. Appellants assume the Examiner intended to reject claim 22 over Czerwinski in view of Johnson.

Third Ground of Rejection

Claim 40 stands finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Czerwinski in view of Appellants' Applied Prior Art. Appellants traverse this rejection for at least the reasons given above regarding its independent claim.

VIII. CONCLUSION

For the foregoing reasons, it is submitted that the Examiner's rejection of claims 1-6, 8-31, 33-47 and 49-72 was erroneous, and reversal of his decision is respectfully requested.

The Commissioner is authorized to charge the appeal brief fee of $500.00 and any other fees that may be due to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5181-64800/RCK. This Appeal Brief is submitted with a return receipt postcard.

Respectfully submitted,

Robert C. Kowert
Reg. No. 39,255
Attorney for Appellants

Meyertons, Hood, Kivlin, Kowert & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
(512) 853-8850

Date: August 1, 2005

IX. CLAIMS APPENDIX

The claims on appeal are as follows.

1. A method for communicating in a distributed computing environment, comprising:

a client accessing an authentication service to obtain an authentication credential to use a first service;

determining client capabilities for said client, wherein said client capabilities are capabilities of said first service that said client is permitted to use;

binding said client capabilities to said authentication credential;

said client sending a first message to said first service, wherein said first message includes said authentication credential;

said first service using said authentication service to authenticate said authentication credential received in said first message; and

said first service responding to said first message if said authentication credential in said first message is determined to be authentic as from said client.

2. The method as recited in claim 1, further comprising said client obtaining an address for said authentication service from an advertisement for said first service, wherein said accessing an authentication service comprises said client sending a message to said address for said authentication service requesting said authentication credential to use said advertised first service.

3. The method as recited in claim 2, wherein said advertisement for said first service includes a data representation language schema defining a message interface for accessing said first service.

4. The method as recited in claim 3, wherein said first message corresponds to a message defined in said data representation language schema.

5. The method as recited in claim 4, further comprising said client sending additional messages to said first service to use said first service, wherein said authentication credential is included with each one of said additional messages, and wherein each one of said additional messages is defined by said data representation language schema.

6. The method as recited in claim 5, wherein said data representation language schema is an eXtensible Markup Language (XML) schema.

8. The method as recited in claim 1, further comprising:

said client sending a request message to said first service to access a capability of said first service, wherein said request message includes said authentication credential;

said first service determining that the capability requested in said request message is within said client capabilities; and

said first service fulfilling said request message only if the capability requested in said request message is within said client capabilities.

9. The method as recited in claim 1, wherein said determining client capabilities comprises said client accessing an access policy service to obtain a capability token indicating which capabilities of said first service said client is permitted to access.

10. The method as recited in claim 9, wherein said authentication service and said access policy service are combined as a single service and wherein said capability token is included within said authentication credential.

11. The method as recited in claim 1, wherein said determining client capabilities is performed by said first service.

12. The method as recited in claim 1, further comprising said client generating a message gate for accessing said first service, wherein said message gate sends request messages from said client to said first service to access said first service, and wherein said message gate includes said authentication credential in each message to said first service.

13. The method as recited in claim 12, further comprising said client obtaining a service advertisement for said first service before accessing said first service, wherein said service advertisement comprises an address for said authentication service and an address for said first service.

14. The method as recited in claim 13, wherein said service advertisement further comprises a data representation language schema defining a message interface for accessing said first service, wherein said message gate verifies that each message sent from said client to said first service complies with said data representation language schema.

15. The method as recited in claim 1, wherein said authentication service is a separately addressable service from said first service.

16. The method as recited in claim 1, wherein said client accessing an authentication service to obtain an authentication credential to use a first service

comprises said authentication service returning said authentication credential to said client only if said client is authorized to access said first service.

17. A method for communication in a distributed computing environment, comprising:

a client obtaining a service advertisement for a first service, wherein said service advertisement includes an address for an authentication service;

said client sending a request message to said authentication service to obtain an authentication credential to use said first service;

said client generating a message gate for accessing said first service, wherein said message gate embeds said authentication credential in every message from said client to said first service; and

said client accessing said first service through said message gate.

18. The method as recited in claim 17, wherein said service advertisement further comprises a data representation language schema defining a message interface for accessing said first service, the method further comprising said message gate verifying that every message sent from said client to said first service complies with said data representation language schema.

19. The method as recited in claim 18, wherein said data representation language schema is an eXtensible Markup Language (XML) schema and said messages from said client to said first service are XML messages.

20. The method as recited in claim 17, further comprising said first service using said authentication service to determine if said authentication credential received in a first message from said client is authentic.

21. The method as recited in claim 20, further comprising, after authenticating said authentication credential received in said first message from said client, said first service determining which capabilities of said first service said client is authorized to use, wherein said first service responds to a request message from said client only if said request message is for an authorized capability for said client.

22. The method as recited in claim 21, further comprising said first service binding a determination of which capabilities of said first service said client is authorized to use to said authentication credential so that said first service does not need to repeat said determining which capabilities of said first service said client is authorized to use.

23. The method as recited in claim 20, further comprising said first service noting whether or not said authentication credential is authentic so that said first service does not need to repeat said using said authentication service to determine if said authentication credential received in a first message from said client is authentic.

24. The method as recited in claim 17, wherein said service advertisement for said first service further includes an address for accessing said first service, wherein said authentication service and said first service are separate services within the distributed computing environment.

25. The method as recited in claim 17, wherein said service advertisement further includes a service identifier token for said first service, wherein said client sending a request message to said authentication service to obtain an authentication credential comprises sending said service identifier token and a client identifier token to said authentication service.

26. The method as recited in claim 25, wherein said authentication service generates said authentication credential from said client identifier token and said service identifier token.

27. A client device configured to:

access an authentication service to obtain an authentication credential to use a first service;

determine client capabilities for said client device, wherein said client capabilities are capabilities of said first service that said client device is permitted to use; and

bind said client capabilities to said authentication credential;

send a first message to said first service, wherein said first message includes said authentication credential, wherein said first service is configured to use said authentication service to authenticate said authentication credential received in said first message; and

receive a response to said first message from said first service if said authentication credential in said first message is determined to be authentic as from said client device.

28. The client device as recited in claim 27, further configured to:

obtain an address for said authentication service from an advertisement for said first service;

wherein, in said accessing an authentication service, the client device is further configured to:

send a message to said address for said authentication service requesting said authentication credential to use said advertised first service.

29. The client device as recited in claim 28, wherein said advertisement for said first service includes a data representation language schema defining a message interface for accessing said first service, and wherein said first message corresponds to a message defined in said data representation language schema.

30. The client device as recited in claim 29, further configured to send additional messages to said first service to use said first service, wherein said authentication credential is included with each one of said additional messages, and wherein each one of said additional messages is defined by said data representation language schema.

31. The client device as recited in claim 29, wherein said data representation language schema is an eXtensible Markup Language (XML) schema.

33. The client device as recited in claim 27, further configured to:

send a request message to said first service to access a capability of said first service, wherein said request message includes said authentication credential;

wherein said first service is configured to fulfill said request message only if said first service determines that the capability requested in said request message is within said client capabilities.

34. The client device as recited in claim 27, wherein, in said determining client capabilities, the client device is further configured to access an access policy service to obtain a capability token indicating which capabilities of said first service said client is permitted to access.

35. The client device as recited in claim 34, wherein said authentication service and said access policy service are combined as a single service, and wherein said capability token is included within said authentication credential.

36. The client device as recited in claim 27, further configured to generate a message gate for accessing said first service, wherein said message gate sends request messages from said client to said first service to access said first service, and wherein said message gate includes said authentication credential in each message to said first service.

37. The client device as recited in claim 36, further configured to:

obtain a service advertisement for said first service before accessing said first service, wherein said service advertisement comprises a data representation language schema defining a message interface for accessing said first service;

wherein said message gate is configured to verify that each message sent from said client device to said first service complies with said data representation language schema.

38. The client device as recited in claim 27, wherein, in said accessing an authentication service to obtain an authentication credential to use a first service, the client device is further configured to receive from said authentication service said authentication credential only if said client device is authorized to access said first service.

39. The client device as recited in claim 27, wherein said authentication service and said first service are configured to execute within a service device, and wherein said client device is further configured to couple to said service device via a network.

40. The client device as recited in claim 27, wherein said client device is further configured to couple to a network via a wireless connection.

41. The client device as recited in claim 27,

wherein said authentication service is configured to execute within an authentication server;

wherein said first service is configured to execute within a service device; and

wherein said client device, said service device, and said authentication server are separate devices comprised in a distributed computing environment.

42. The client device as recited in claim 27, wherein said first service is configured to execute within said client device.

43. A service device configured to:

receive from a client a first message including an authentication credential, wherein said client accesses an authentication service to obtain said authentication credential to use said service device;

use said authentication service to authenticate said authentication credential received in said first message;

determine client capabilities for said client, wherein said client capabilities are capabilities of said service device that said client is permitted to use;

bind said client capabilities to said authentication credential; and

respond to said first message if said authentication credential in said first message is determined to be authentic as from said client.

44. The service device as recited in claim 43, further configured to provide to said client an advertisement for said service device, wherein said advertisement includes a data representation language schema defining a message interface for accessing said service device.

45. The service device as recited in claim 44, wherein said first message corresponds to a message defined in said data representation language schema.

46. The service device as recited in claim 45, further configured to receive additional messages from said client to use said service device, wherein said authentication credential is included with each one of said additional messages, and wherein each one of said additional messages is defined by said data representation language schema.

47. The service device as recited in claim 44, wherein said data representation language schema is an eXtensible Markup Language (XML) schema.

49. The service device as recited in claim 43, further configured to:

receive from said client a request message to access a capability of said service device, wherein said request message includes said authentication credential;

determine that the capability requested in said request message is within said client capabilities; and

fulfill said request message only if the capability requested in said request message is within said client capabilities.

50. The service device as recited in claim 43, wherein said client is configured to execute within a client device, and wherein said service device and said client device are separate devices comprised in a distributed computing environment.

51. A distributed computing system, comprising:

a client device; and

a service device;

wherein said client device is configured to:

access an authentication service to obtain an authentication credential to use said service device;

determine client capabilities for said client device, wherein said client capabilities are capabilities of said service device that said client device is permitted to use;

bind said client capabilities to said authentication credential;

send a first message to said service device, wherein said first message includes said authentication credential; and

wherein said service device is configured to:

use said authentication service to authenticate said authentication credential received in said first message; and

respond to said first message if said authentication credential in said first message is determined to be authentic as from said client.

52. The system as recited in claim 51,

wherein the service device is further configured to provide to said client device an advertisement for said service device, wherein said advertisement includes a data representation language schema defining a message interface for accessing said service device;

wherein the client device is further configured to obtain an address for said authentication service from said advertisement for said service device; and

wherein, in said accessing an authentication service, the client device is further configured to send a message to said address for said authentication service requesting said authentication credential to use said advertised service device.

53. The system as recited in claim 52, wherein said advertisement for said service device includes a data representation language schema defining a message interface for accessing said service device, wherein said first message corresponds to a message defined in said data representation language schema.

54. The system as recited in claim 53, wherein the client device is further configured to send additional messages to said service device to use said service device, wherein said authentication credential is included with each one of said additional messages, and wherein each one of said additional messages is defined by said data representation language schema.

55. The system as recited in claim 53, wherein said data representation language schema is an eXtensible Markup Language (XML) schema.

56. The system as recited in claim 51, wherein said authentication service is configured to execute within said service device.

57. The system as recited in claim 51,

wherein said authentication service is configured to execute within an authentication server; and

wherein said client device, said service device, and said authentication server are separate devices comprised in a distributed computing environment.

58. A distributed computing system, comprising:

a client device;

a service device;

wherein said client device is configured to:

obtain a service advertisement for said service device, wherein said service advertisement includes an address for an authentication service;

send a request message to said authentication service to obtain an authentication credential to use said service device;

generate a message gate for accessing said service device, wherein said message gate is configured to embed said authentication credential in every message from said client device to said service device; and

access said service device through said message gate;

59. The system as recited in claim 58,

wherein said service advertisement further comprises a data representation language schema defining a message interface for accessing said service device; and

wherein said message gate is further configured to verify that every message sent from said client device to said service device complies with said data representation language schema.

60. The system as recited in claim 59, wherein said data representation language schema is an eXtensible Markup Language (XML) schema and said messages from said client device to said service device are XML messages.

61. The system as recited in claim 58, wherein said service device is configured to:

use said authentication service to determine if said authentication credential received in a first message from said client device is authentic;

determine which capabilities of said service device said client device is authorized to use; and

respond to said first message from said client device only if said first message is for an authorized capability for said client device.

62. A carrier medium comprising program instructions, wherein the program instructions are computer-executable to implement:

a client accessing an authentication service to obtain an authentication credential to use a first service;

determining client capabilities for said client, wherein said client capabilities are capabilities of said first service that said client is permitted to use;

binding said client capabilities to said authentication credential;

said client sending a first message to said first service, wherein said first message includes said authentication credential;

said first service using said authentication service to authenticate said authentication credential received in said first message; and

said first service responding to said first message if said authentication credential in said first message is determined to be authentic as from said client.

63. The carrier medium as recited in claim 62, wherein the program instructions are further computer-executable to implement:

said client obtaining an address for said authentication service from an advertisement for said first service;

wherein, in said accessing an authentication service, the program instructions are further computer-executable to implement:

said client sending a message to said address for said authentication service requesting said authentication credential to use said advertised first service.

64. The carrier medium as recited in claim 63, wherein said advertisement for said first service includes a data representation language schema defining a message interface for accessing said first service, wherein said first message corresponds to a message defined in said data representation language schema.

65. The carrier medium as recited in claim 64, wherein said data representation language schema is an eXtensible Markup Language (XML) schema.

66. The carrier medium as recited in claim 62, wherein the program instructions are further computer-executable to implement:

said client sending a request message to said first service to access a capability of said first service, wherein said request message includes said authentication credential;

said first service determining that the capability requested in said request message is within said client capabilities; and

said first service fulfilling said request message only if the capability requested in said request message is within said client capabilities.

67. The carrier medium as recited in claim 62, wherein the program instructions are further computer-executable to implement:

said client generating a message gate for accessing said first service;

said message gate sending request messages from said client to said first service to access said first service, wherein said message gate includes said authentication credential in each message to said first service.

68. The carrier medium as recited in claim 67, wherein the program instructions are further computer-executable to implement:

said message gate verifying that each message sent from said client to said first service complies with a data representation language schema, wherein said data representation language schema defines a message interface for accessing said first service

69. A carrier medium comprising program instructions, wherein the program instructions are computer-executable to implement:

a client obtaining a service advertisement for a first service, wherein said service advertisement includes an address for an authentication service;

said client sending a request message to said authentication service to obtain an authentication credential to use said first service;

said client generating a message gate for accessing said first service, wherein said message gate embeds said authentication credential in every message from said client to said first service; and

said client accessing said first service through said message gate.

70. The carrier medium as recited in claim 69, wherein said service advertisement further comprises a data representation language schema defining a message interface for accessing said first service, and wherein the program instructions are further computer-executable to implement:

said message gate verifying that every message sent from said client to said first service complies with said data representation language schema.

71. The carrier medium as recited in claim 70, wherein said data representation language schema is an eXtensible Markup Language (XML) schema and said messages from said client to said first service are XML messages.

72. The carrier medium as recited in claim 69, wherein the program instructions are further computer-executable to implement:

said first service using said authentication service to determine if said authentication credential received in a first message from said client is authentic;

said first service determining which capabilities of said first service said client is authorized to use; and

said first service responding to said first message from said client only if said first message is for an authorized capability for said client.

X. EVIDENCE APPENDIX

No evidence submitted under 37 CFR §§ 1.130, 1.131 or 1.132 or otherwise entered by the Examiner is relied upon in this appeal.

XI. RELATED PROCEEDINGS APPENDIX

No decision has yet been rendered in the related appeal identified above.